

**INFORMATIQUE
NIVEAU SUPÉRIEUR
ÉPREUVE 2**

Mardi 18 mai 2004 (matin)

2 heures 30 minutes

INSTRUCTIONS DESTINÉES AUX CANDIDATS

- N'ouvrez pas cette épreuve avant d'y être autorisé.
- Répondez à toutes les questions.

Page blanche

1. Un système flexible robotisé stocke ses tâches dans une file d'attente. Chaque tâche est dotée d'un nom de fichier et d'un numéro de priorité situé entre 0 et 7.

- (a) Définissez une structure d'enregistrements adaptée au stockage d'une tâche.

[3 points]

La file d'attente est un tableau de 8 pointeurs vers les listes liées d'enregistrements. La liste liée vers laquelle pointe l'élément 0 du tableau a la priorité la plus élevée. L'élément 7 pointe vers la liste qui a la priorité la plus basse. Le tableau de pointeurs est intitulé `TACHE_Q` et est global. `TACHE_Q` est initialisé avec des valeurs `nil`.

Tableau de pointeurs

0
1
2
3
4
5
6
7

Listes liées d'enregistrements

⇒ enregistrement ⇒ enregistrement

⇒ enregistrement

⇒ enregistrement ⇒ enregistrement ⇒ enregistrement

⇒ enregistrement

- (b) Définissez le tableau et une nouvelle structure d'enregistrements, `TACHE`, qui seraient adaptés aux enregistrements des listes liées.

[2 points]

- (c) Construisez une **procédure** `AJOUTER_TACHE(val J pointer -> TACHE)` qui ajoute la tâche à la fin de la file d'attente pertinente. Supposez que `TACHE` a été créée et que ses valeurs ont été saisies.

[11 points]

- (d) Une fonction est requise pour supprimer la tâche au plus haut niveau de priorité, et ce, depuis le plus longtemps. L'algorithme doit également renvoyer un pointeur à cette tâche ou un pointeur `nil` si la file d'attente est vide. Construisez l'algorithme de ce programme. Appelez-le `SUPPRIMER_TACHE`.

[11 points]

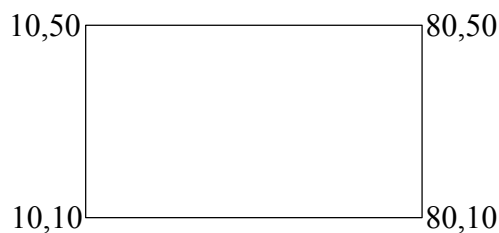
Un nouvel algorithme est envisagé pour tester la file d'attente chaque fois qu'elle est consultée. L'algorithme testera chaque tâche pour vérifier si elle se trouve dans la file d'attente, à sa priorité actuelle, depuis plus de 10 minutes. Si c'est le cas, elle est remontée d'un niveau de priorité.

- (e) À l'aide de `AJOUTER_TACHE`, décrivez l'algorithme que vous écririez pour réaliser cette opération.

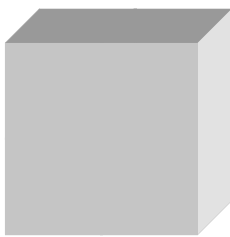
[3 points]

Cette question nécessite la lecture de l'étude de cas.

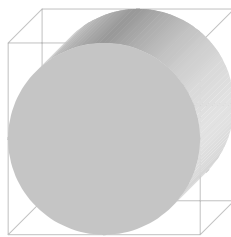
2. (a) Exposez **trois** avantages de l'utilisation d'ordinateurs et d'un équipement informatique dans le processus de fabrication. [6 points]
- (b) Comparez les approches variante et générative de la programmation pièce pour la commande numérique. [4 points]
- (c) Écrivez le programme pièce pour la commande numérique afin de construire la forme suivante. On suppose un outil de découpe de 10 mm de diamètre. L'outil se trouve actuellement en bas, à gauche à la position $X = 0, Y = 0$. [5 points]



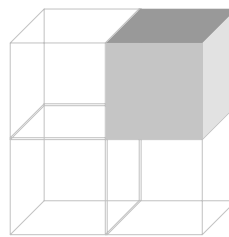
- (d) Dessinez l'objet obtenu à partir des primitives suivantes (A, B, C et D) selon l'arbre binaire en-ordre. [3 points]



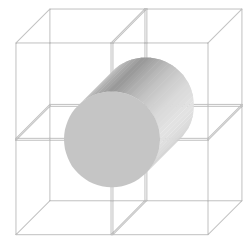
A



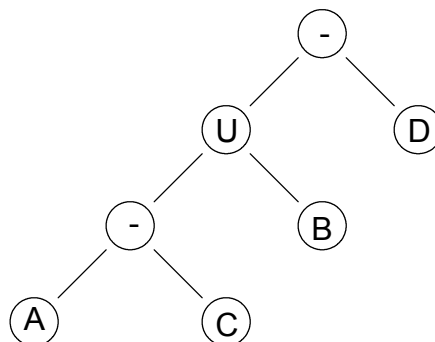
B



C



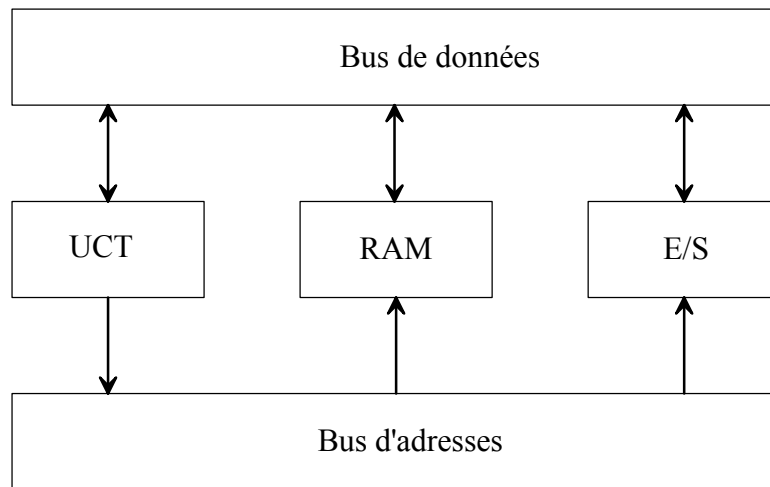
D



- (e) Exposez **trois** raisons expliquant pourquoi le traitement distribué est utilisé à différents niveaux dans un grand réseau d'usine. [6 points]
- (f) Comparez les modélisations fil de fer, surfacique et volumique par rapport à leur pertinence dans la modélisation d'assemblage. [6 points]

3. Un capteur de tension est fixé à l'outil de découpe d'une machine. Il génère une petite tension électrique qui est proportionnelle à la quantité de tension à laquelle la machine est soumise. Cette dernière est contrôlée par un ordinateur qui vérifie la quantité de tension à laquelle l'outil est soumis et éteint le moteur de la machine si elle devient excessive.

- (a) Dessinez un ordinogramme de système du processus en représentant le capteur de tension, l'ordinateur et la machine. [5 points]
- (b) À l'aide du diagramme ci-dessous, expliquez comment le bus de données et le bus d'adresses fonctionnent ensemble pour communiquer avec la machine et le capteur. [3 points]



- (c) Définissez ce qu'est une *entrée/sortie* (E/S) à *topographie mémoire* dans ce contexte. [4 points]
- (d) Expliquez comment la mémoire cache pourrait être utilisée pour améliorer la performance de ce système. [3 points]

4. L'algorithme suivant supprime d'un tableau les valeurs en double à l'aide d'un tableau temporaire qui est ensuite recopié dans l'original.

```

declare N, I, J, PLA integer
declare DOUBLE boolean
declare DONNEES, TEMP integer array [1..N]

/*DONNEES et N sont initialisées hors de l'algorithme */

PLA <-- 1

for I <-- 1 upto N do
    DOUBLE <-- faux
    for J <-- 1 upto N do
        if DONNEES[I] = TEMP[J] then
            DOUBLE = vrai
        endif
    endfor
    if not DOUBLE then
        TEMP[PLA] <-- DONNEES[I]
        PLA <-- PLA + 1
    endif
endfor

for I <-- 1 upto N do
    if I >= PLA then
        DONNEES[I] <-- 0
    else
        DONNEES[I] <-- TEMP[I]
    endif
endfor

```

- (a) Indiquez l'efficacité grand-O de l'algorithme. [2 points]
- (b) Expliquez les améliorations qui pourraient être réalisées afin de rendre l'algorithme plus efficace en matière d'utilisation de la mémoire ou de temps d'exécution. [8 points]

(Suite de la question à la page suivante)

(Suite de la question 4)

Soit l'algorithme suivant qui implémente un arbre de décision.

```

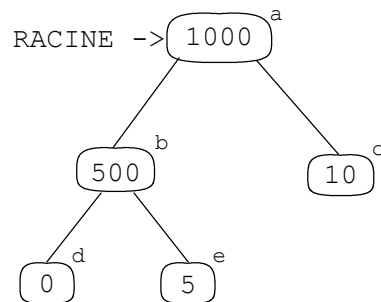
newtype NOEUD record
  A integer
  GAUCHE pointer -> NOEUD
  DROITE pointer -> NOEUD
endrecord

function REMISE (ref N pointer -> NOEUD, val X integer)
  result integer
  if not ((N -> GAUCHE = nil) and (N -> DROITE = nil)) then
    if X > N -> A then
      return REMISE (N -> DROITE, X)
    else
      return REMISE (N -> GAUCHE, X)
    endif
  else
    return N -> A
  endif
endfunction REMISE

```

- (c) Si l'arbre est défini comme suit, tracez l'algorithme de l'appel
 REMISE (RACINE, 575).

[5 points]



Les cercles montrent la valeur de A dans chaque nœud de l'arborescence.

5. Une base de données de très grande taille et contenant des enregistrements non triés et non indexés est stockée dans un ordinateur.

(a) Indiquez le type de structure de fichiers utilisé. *[1 point]*

(b) Décrivez **un** avantage et **un** inconvénient de ce type de stockage des données. *[4 points]*

Parfois, une impression triée des données est nécessaire. La base de données est de trop grande taille pour la mémoire primaire.

(c) Exposez le type ou le tri qui serait nécessaire pour organiser les données afin d'obtenir une impression triée et justifiez votre réponse. *[3 points]*

La base de données ne doit pas être triée, mais un meilleur accès et une fonction de tri sont nécessaires.

(d) Indiquez une autre structure de fichiers qui permettrait d'obtenir ce résultat et expliquez pourquoi elle serait adaptée. *[3 points]*

(e) Décrivez pourquoi cette structure de fichiers faciliterait certaines opérations, comme le tri. *[4 points]*